Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 1730 – 1734

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

Research on a New Method of Index in Three-Dimensional Space

Yongshan Liu*,Xiaoxia Wang

College of Information Science and EngineeringYanshan University, Qinhuangdao Hebei 066004, P.R.China

Abstract

In this paper we propose a CTR-tree index and corresponding operation algorithms. We divide objects into some clusters using k-means algorithm, and then establishes CTR-tree structure with 3D topology restriction for those clusters. The experimental results show that t the size of the nodes tends to be uniform by reducing overlap volume among adjacent nodes and it improves the search efficiency greatly.

Key words: Spatial Index; Topological Constraint rule, Spatial Cluster Grouping; 3D GIS;

1. Subject background

Because of maturity and consummation of theory and technology in 2D database, and continuous development of computer technology, as well as the strongly demands of application, 3D spatial database has already been a new hot spot. Currently, the studies of 3D spatial index mostly concentrated on Octree and 3D R-tree. These indexes in 3D-GIS have their proper scope, if not making a distinction these indexes will show many shortcomings and be inefficient in dealing with large amount of data. Literature [1] points out that distribution of spatial objects has to be predicted before Octree construction, and when dealing with a large amount of data, it is difficult to hold the level. For objects with longer life cycle or greatly changed location, 3D R-tree index efficiency may be reduced because of increased overlap of 3D

* Corresponding author. Tel.: +86-0335-8501977; fax: +86-0335-8500223.
E-mail address: wangxiaoxia10-11@163.com.

doi:10.1016/j.proeng.2012.01.203

MBB (minimum boundary box)[2]. Due to the existing defect of the above index structure, studying a new type of 3D index has important theoretical and practical application value in meeting user's demands.

2. CTR-Tree

We created a new 3D index named CTR-tree based on classical R-tree. First we divide objects into groups with the algorithm of 3D spatial clustering, and then constrained each group with 3D spatial topology. The Specific process to create the CTR-tree is described through the following two parts.

2.1. The 3D Cluster Grouping

To create CTR-tree, the first step is the 3D cluster grouping. Referring to literature [3], dividing the space into k cluster groups by k-means algorithm of clustering to make spatial adjacent objects stored in the same or adjacent nodes, which makes the shape of the sibling nodes more reasonable and the size of them more evenly. As shown in figure 2, obviously, it's more reasonable to divide the wire-frame cube into three subgroups than two. According to the Gauss formula mentioned in Literature [4], the formula of [(*Xi* + *Yi* + *Zi*) / 3] *3 > = *Xi* * *Yi* * *Zi* is established if and only if the equation *Xi* = *Yi* = *Zi* establishes, that is to say the more smaller of the node, the more closer to the cube of the shape.

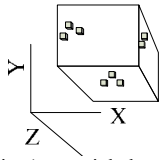

Fig. 1. spacial cluster grouping

In the spatial cluster grouping, we should first determine the maximum number of the groups kmax, which is the maximum value of k that can make N / k > =m. The value of k which can make the sum volume of spatial coverage and overlap W (in formulae 1) minimum is seemed as the final number of the clusters.

$$W = \sum_{i=1}^{n} \mathrm{Coverage}(V_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{Overlap}(V_i, V_j) \quad (1)$$

Execute the following algorithms respectively to group the objects from group number $k = 2$ to $k = kmax$.

Input: the 3D spatial objects set $S = \{O1, O2, \ldots, On\}$

Output: the objects set Si($i = 1, \ldots, k$) after grouping

- Selected k objects from the set S as seeds to make sure they have the greatest distances, and let the centers of the seed objects as centers of the k small collections respectively.
- Assign each object to the group with the closest centroid, namely to make follow Euclidean distance $d = [\,[xi-x]2+[yi-y]2+[zi-z]2]^{\frac{1}{2}}$ minimum.
- Recalculate the positions of the K centroid after adding new objects to Si
- Jump to step(2), repeating the rest until the object are totally distributed to the k sets.
- Finally, sort the centers of the groups, and stored them in ordered chain list.

2.2. The constraints rules of 3D spatial topology to construct CTR-tree

Secondly, we construct tree structure for each group appended with 3D spatial topology. We consider objects that touch each other as a spatial partition, and disjoint objects (i.e. $A \cap B = \Phi$) for another spatial

partition. Referencing to literature [5], constraint rules that CTR tree used are as follows (not suit large linear or planar objects such as rivers, cadastre, unless do same cutting treatment towards them).

- For single spatial object whose 3D scale ratio R exceeds the threshold value T, we split it into several spatial objects with suitable 3D scale ratio in the exceeded dimension (as shown in figure2 (A)).
- For objects whose 3D spatial stretching ratio in different coordinate axis exceeds prescribed threshold value T and $A \cap B \neq \Phi$, we regard each of them as one node respectively (as shown in Figure 2 (B).
- For objects whose 3D scale ratio R within the scope of threshold value T and $A \cap B \neq \Phi$, we use the same external hexahedron to share the same parent node (as shown in Figure 2(C)) Furthermore, each spatial object links in the way of chain list.

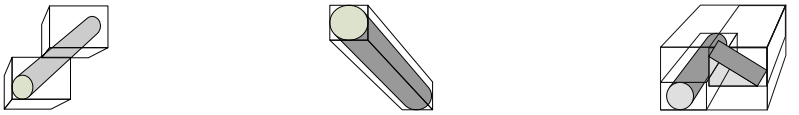

Fig. 2. (a) Splitted Spatial Objects; (b) Ssingle Spatial Object Intersectant ; (c)Spatial object

2.3. The structure of CTR-tree

The structure of CTR - tree is different from that of 3 D R-tree. The cluster MBB (Minimum Bounding Box) of CTR - tree preserves the pointers of both next cluster MBB and children nodes, and is organized according to 3D spatial topology constraint rules in every cluster group. As shown in Figure 3, the output objects are divided into K clusters *(G1*, *G2 ... Gn*) and sorted according to the coordinates of the centers. Cluster MBB are preserved in the form of orderly list and the time complexity of establishing list is O (K).

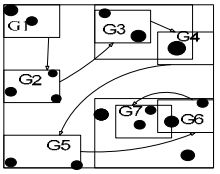

Fig.3. Plane Graph of CTR-tree Structure.

2.4. Search algorithm

Firstly we compare MBB of referenced objects to spatial cluster centres to see which group are search terms in, that is to determine the cluster MBB of search terms with time complexity of O (log$K$).

Algorithm SearchNode(*root*,*P*)
Input: *root* is the root of current CTR-tree; *P* is the data we need to query
Output: *Adr* is the physical address of the object *P*
/* Compare the query object *P* with the spatial cluster centers to find its possible subgroup MBB*/
1. For i=1 to K do
2. *G*=FindGroupMBR(*P*) // Return the grouping node MBB of *P*
3. If(*G* not empty) // If the cluster group node of *P* not empty, then continue
4. If(P inside G or P overlap G) // If P inside or overlap G, continue to search
/*Call the search algorithms recursively in the descendant nodes until it reaches the leaf nodes.*/
5. For each child node g in *G*
6. SearchNode(*g*,*P*)
7. If(find)

Return adr // If find, then return the physical address adr.
End Algorithm SearchNode

2.5. Insert algorithm

To insert node *O* we first iteratively find insert position and parent *P* of *O*. If there are vacancies in *P*, record MBB and ID of *O*, otherwise call for cell splitting, then adjust tree structure. Given prescribed threshold value *T* and calculate 3D scale ratio *R* of *O* and topological relation *tp* between *O* and others.

Algorithm InsertNode(*Record*, *P*)
Input: *Record* is the record to insert
Output: CTR-tree with new records inserted
/* Invoke search algorithm to find the subtree *N* where the new records should be inserted into*/
1. *N*=SearchNode(*root*, *Record*)
2. If(*R*>*T*) // If the 3D scale ratio *R* of *P* exceeds the threshold value *T*, then split
3. SplitP() // Split *P* into objects P1, *P2* ...with same attribute according to topological rule 1
4. InsertNode(*P1*,*P2*……) // Invoke insert algorithm to insert objects *P1*, *P2* ... into CTR-tree
/*According to topological rule 2, if intersection, find all intersecting spatial objects and merged them into the same node which named *asOneNode**/
5. Else If(*tp* == overlap) *asOneNod*=Findalloverlap ()
/*If all the objects satisfy the threshold conditions, insert the whole as an independent node*/
6. If(*qr*<*T*) Insert (*N*, *asOneNode*)
/* If the 3D scaling ratio exceeds the threshold, then insert into Leaf nodes as a separate node*/
7. Else InsertP(*N*, *Pqs*)
/* According to the topological rule 3, if separated, insert each of them as separate node*/
8 .Else If(*tp* == disjoint)
InsertP()
9. If(Overflow)
10. OverflowTreatment() //If overflow invoke OverflowTreatment to adjust it
End Algorithm InsertNode

2.6. Splitting algorithm

The inserted node needs to be split when node number exceeds the upper limit of the index record. Splitting algorithm of CTR-tree shows how to properly divide a rectangular box set into smaller rectangular box sets. When the insertion of new index item causes overflow of nodes, we use C-Linear splitting algorithm combined with k-means algorithm using two-way splitting scheme for node splitting.

2.7. Delete algorithm

For object deletion, first we use searching algorithm to find the leaf index nodes of the spatial object *O* to be deleted, then remove it, and finally adjust tree structure. Searching from root of the tree, if the MBB of a nonleaf node contains the MBB of *P*, continue searching until leaf node. Dropping of index items may cause leaf nodes underflow, remove it and reinsert the rest of the index items to adjust the tree.

3. The experimental results and analysis

The execution time is decided by I / O access number and CPU comparison time, so we use I / O

access time as performance metric here. 3200 and 3500 randomly generated MBB data sets are used as the object sets. Through experimental results shown in Fig.4 we can find that cluster grouping makes adjacent objects stored in the same or adjacent node and 3D spatial topological constraint rules reduces range overlap among sibling nodes to improve utilization rate of nodes. The insert algorithm makes overlap of sibling nodes improved significantly, and redundancy operation such as multi-path query is decrease quickly. This research enhanced the efficiency of spatial query by more than 30%.

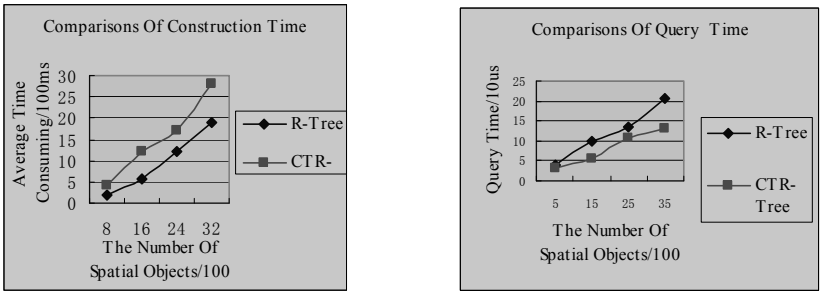

Fig. 4. (a) Comparisons of Construction Time; (b) Comparisons of Query Time

4. Conclusion

3D spatial index is the premise of realizing 3D spatial data efficient management and real-time applications. To overcome defects of serious overlap among nodes and uneven node size in3D direction of R-tree, we use spatial cluster grouping to construct CTR-tree appended with 3D topological constraint. The experimental result shows superiority of the research method in improving performance of search efficiency. Next step of the research work will be focused on application of this method in higher dimensional data and further uses of hybrid 3D data model to make3D topological constraint rules more perfect. Then we can study more efficient high dimensional CTR-family based on this research.

Acknowledgements

This work is supported in part by the Natural Science Foundation of Hebei Province P.R. China under Grant No.F2009000473; the 2008 Electron Information Industry Development Fund of Ministry of Information Industry under Grant No.[2008]97.

References

[1] LIU Yan, MA Jing-song, ZHANG Yong-yu. Studies on R-tree spatial index for 3D GIS. Science of Surveying and Mapping;2010.

[2] XIAOHui1, LIQing-quan. Access methods in moving objects databases. Journal ofComputerApplication;2010

[3] ZHU Qing, GONG Jun. An Improved Full 3D R2Tree Spatial Index Method. Geomatics and Information Science of Wuhan University;2006

[4] WANG Jing, JIANG Gang-wu, GUO Rui. Qualitative Detailed Description for Spatial direction relations.Remote Sensing and Spatial Information sciences;2008:32(B2).

[5] CHEN Peng, MENG Lingkui, SONG Yang. R-tree Structure Appended with Spatial Topology Restrictions in3D GIS. Geomatics and Information Science of Wuhan University;2007.